Bringing the Semantic Web home:
a research agenda for local, personalized SWUI

Paul André
Electronics and Computer Science
University of Southampton, UK
pa2@ecs.soton.ac.uk

m.c. schraefel
Electronics and Computer Science
University of Southampton, UK
mc@ecs.soton.ac.uk

Max Van Kleek
MIT CSAIL
Cambridge, MA, USA
emax@csail.mit.edu

David Karger
MIT CSAIL
Cambridge, MA, USA
karger@mit.edu

CHI 2008, April 5 – April 10, 2008, Florence, Italy
ACM 1-xxxxxxxxxxxxxxxxx.

Abstract
We suggest that by taking the Semantic Web local and personal, and deploying it as a shared "data sea" for all applications to trawl, new types of interaction are possible (even necessitated) with this heterogeneous source integration. We present a motivating scenario to foreground the kind of interaction we envision as possible, and outline a series of associated questions about data integration issues, and in particular about the interaction challenges fostered by these new possibilities. We sketch out some early approaches to these questions, but our goal is to identify a wider field of questions for the SWUI community in considering the implications of a local/social semantic web, not just a public one, for interaction.

Keywords
semantic web, structure, user interaction, personalized

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
A rationale for the SWUI community has been that with structured data in the wild, at web-scale, there may be new types of interaction challenges that current interaction paradigms (dealing with 'Web1.0/2.0', applications and their specific data and well-formulated tasks) have not been designed for. So far, work in this

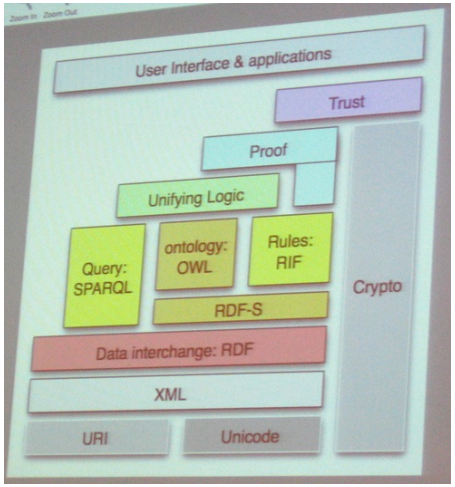

figure 1. Semantic Web Layer Cake (with user interface layer) as presented by Tim Berners-Lee. (http://flickr.com/photos/pshab/291147522)

young community has largely focused on integrating heterogeneous data sources on the Web (mashup makers), or on locally controlled, large(ish) datasources (Longwell[1], mSpace [6]) organised by ontologies and semantic web protocols. The success of our efforts thus far has been embodied in the addition of an interface layer to the Semantic Web Layer Cake (fig.1) presented by Web founder Tim Berners-Lee at SWUI06.

There are still considerable challenges to be addressed in provisioning real-time, responsive UIs that operate across networked data sources that take advantage of the richness of heterogeneous data. Many of these challenges are captured in discussions of exploratory search [10] and what is coming to be known as "information seeking support systems" work. We have recently been looking at pushing the Semantic Web/User Interaction question from a different angle than out there, as it were, on the Web alone. We have been considering what happens when we take the SW personal, and imagine it as the foundation of a person's personal computing/personal information space. We posit a more personal, local SW allows (even necessitates) new types of interaction, resulting in richer value from information than either individual applications or their associated data silos can provide. Though "semantic desktop" work exists [1], we have been considering distinctly different interaction design challenges in generating context as RDF data, and UIs for defining (proactive) contextual actions.

Key to our vision are a number of motivating observations and assumptions:

- users need to keep up to date with an ever-increasing volume of structured data (in the form of RDF and RSS) already on the Web (as well as e-mail, etc.);
- increasingly, individual users are *producing* such data too, in the form of location[2], activities[3], schedule[4], music listening habits[5], and it is possible to obtain even more personal structured information directly from the user's computer [7];
- there are/will be rich sources for information about (other) people, places, and things available on the web available in structured form.

We suggest that challenges which stem from these considerations are roughly centered around a) re-contextualizing and re-use of information, and b) automation of currently many manual information management tasks. Each of these approaches is enabled by structured, semantically annotated data, and each requires fundamental interaction design consideration. In the following sections we sketch out these possibilities and postulate components of a SWUI research agenda for exploring them. Our goal in this workshop presentation is to engage with the SWUI community in refining such an agenda.

Re-finding and Re-use

With a semantic storage layer at the local level, the availability of *even a little* bit of structured information that we can capture and associate automatically means we have potentially many new ways to interrogate our own information. Faceted browsers we have explored in

[1] http://simile.mit.edu/longwell

[2] Plazes – http://plazes.com
[3] Twitter – http://twitter.com
[4] Google Calendar – http://google.com/calendar
[5] Last.fm – http://last.fm

the context of the SW at large can be brought to bear locally to facilitate such questions as "what are the web pages I looked at when I was in meeting X?", or "what are all my notes and meetings related to project Y?". However, these are largely read-only explorers (with the exception of tagging), and these types of UI privilege exploration, not creation or adding in new sources for exploration on the fly. Mechanisms to support identifying content for facet categories need to be explored, as do fundamental models for capturing information itself. It is not clear that the same data-specific applications are always appropriate.

We have begun to explore context capture possibilities [9] and while we have a prototype UI for presenting contexts for reuse and rediscovery, it is in no way complete. Assuming we have the benefits of associated data and semantic information (e.g., time, location, open documents) that a semantic data sea on the desktop gives us, we have new opportunities to explore a variety of visualizations depending on the interest of the person exploring the information. One pertinent area of research is simply to begin to explore optimal segmentation of captured data units for later re-use (e.g. sharing with others, or associating with multiple projects).

Sharing, Awareness and Automation

Once we have structure and rich contexts captured and semantically available, it is possible to reconsider automation that has formally been lain on the graves of Office Assistant Clippy and Microsoft Bob. For instance, in Apple's 1987 Knowledge Navigator vision an agent with a natural language interface manages semantically rich information: able to present the comparison of data from a published paper with similar data from another source, or to bring up lecture information from one year and contact the person's acquaintances who research in that area. In the late 1980s such a vision was seen largely as an impossible mission for machine learning. In the early 21st Century, for SW researchers, this kind of data integration and interaction is exactly the kind of support informing its development [2].

The anticipated problem for any kind of machine-assisted interaction is that it is too difficult to get the interaction right. We hypothesize that with structured data we begin to improve the ability of an automation component to provide benefit. Global uniquely identifying URIs for people, places, things, concepts mean searching for additional information about that entity would just involve finding references to that URI. Rather than fixed, absolute global measures; rich, structured relational information allows the computation of derived metrics (similarity, compatibility, suitability) to suit the particular needs of a user, e.g. find a gift similar to product X, but that is less expensive, or from a different vendor.

We have also begun to postulate an approach for considering interaction refinement with such automation by considering a human model: the personal assistant. A personal assistant eases their client's workload by utilising domain and contextual knowledge (e.g. manage calendar, rebalance schedule when needed, prioritize presentation of documents). However, a PA needs an initial discussion with their client to understand expectations, personal habits and preferences, and/or iterative discussions to refine that initial discussion. It would be rare to fire a PA because mistakes occur and exchanges are necessitated to refine the quality of the interaction. We are, in some

cases, satisfied to iterate on tasks. Our interaction challenge is to learn from this model of PA interaction and see what lessons can be defined for designing interactions to support automatic task support. At this point an example of the kinds of automation that might be possible may help explain our thinking.

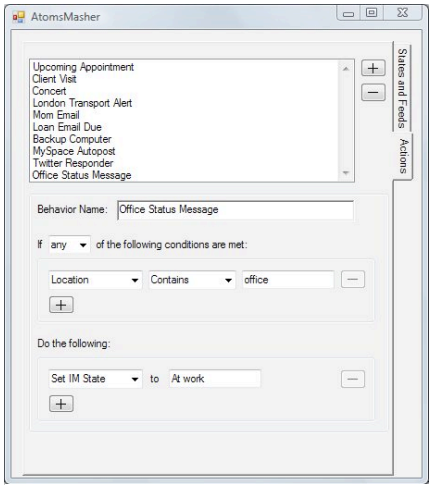

figure 2. AtomsMasher behaviour interface allowing specification of how actions should respond to states and feeds.

Blending Public and Personal Data Sources

To illustrate how a personal assistant (or what we think of as a perfect digital assistant [5]) might help us, we consider information overload tasks: tasks that we would carry out if we had either the time or the inclination. To this end we consider not only personally generated data, but data of personal or social relevance to us, so the sources may exist across domains, for instance: locally (calendar data), socially (music preferences) and publicly (event information).

We started by exploring how semantics enables a system to blend personal and public data: prototyping a tool we call AtomsMasher [8] to use RSS feeds as a context to inform automatic actions. AtomsMasher is able to combine information from heterogeneous sources not just about the person but about things in the world (people, places, books, concerts), and allows people to create rich reactive behaviours. In developing the tool, we have been able to carry out an initial exploration of both the usability and predictability issues of creating such behaviours. These are core research challenges that we will discuss in more detail later, here we present a cutdown scenario illustrating potential usage of AtomsMasher, before focusing on implementation and research challenges.

Yoshiko has a meeting with clients in Rome this afternoon, and as noon approaches the meeting rises to the top of her to-do list. When she leaves for the train station, the location aware software on her mobile phone updates her Plazer RSS location feed. AtomsMasher identifies this action -- leaving the office prior to a scheduled meeting at a different location -- and sets her Away state on her enterprise IM at her company to "Away from the office; en route to Rome for meeting with Interaction Designer", taking location and description fields from the entry in her calendar.

Once on the train, Yoshiko flips open her mobile phone and looks at her personalised newspaper. A concert announcement has appeared on the front page, not recognising the name, she clicks for more information. The newspaper reveals an explanation: the band in question has recently started appearing often on her friends' last.fm feed, that friends' calendar feed shows them free during the concert, and Yoshiko's online calendar shows her to be near the concert venue an hour before.

Fig. 2 shows a behaviour interface allowing specification of how actions should respond to states and feeds in a manner similar to Mail Filters or Smart Playlists. Fig. 3 and 4 show the configuration interface and state table view, representing a characterisation of the user's state, derived from input feeds. Using one or a combination of these views, the user is able to associate a particular feed (friends' last.fm music) with a certain behaviour (notify me of gigs) given a certain condition (band also appears in my last.fm feed).

This tool highlights both back-end challenges in structure, and the interface and interaction issues those challenges raise. Simple schemas (RSS0.95/2.0, ATOM) intended only for news syndication are forced to

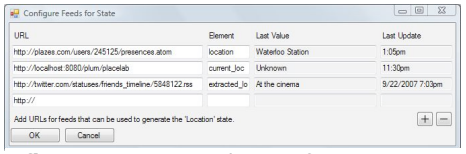

figure 3. AtomsMasher configuration interface, showing feeds for a specific element.

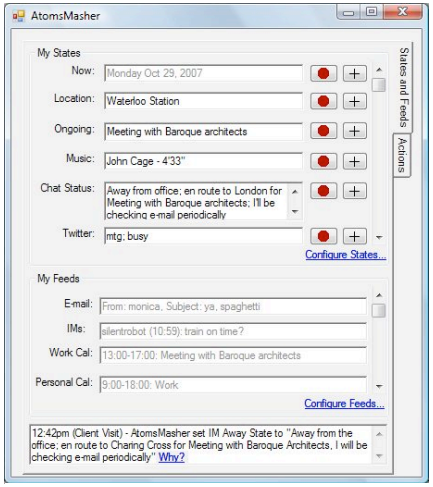

figure 4. AtomsMasher state table and feed view, representing a simple characterization of the user's state, derived from input feeds.

accommodate very different data (e.g. news 'article title' used for music fields of 'track', 'title', and 'artist'), and many websites make arbitrary mappings and 'cram' multiple fields into one in order to retain compatibility with RSS aggregators. In AtomsMasher we applied transformations using perl or Yahoo! Pipes to obtain relevant data, and while this is not beyond the ability of developers, it is beyond what most end-users are willing to attempt. Other websites used external XML schemas (e.g. xCal[6], geo[7]) containing appropriate fields for the data items they were describing. Though more complex, this mitigated the need to write transformation scripts. The interaction challenges are foremost here: allowing the user to manually identify the bits they want out of feeds through the AtomsMasher interface and link them to form rules for reactive behaviours. These problems highlight the need for structure and user tools to utilise these feeds other than passively in RSS aggregators.

Understanding end-user customisation: though we proposed two candidate user interfaces for specifying behaviours in AtomsMasher (see fig. 2,3,4), we are not satisfied that these will fully engage users. For instance, while our more "user friendly" UI for behaviour specification is based on the mail filter paradigm of menu-based selectors, we know that despite their utility very few people use mail filters [3]. Just as mail filters are not required to manage email, AtomsMasher is not required to process Web information. This leads to a bigger question: how many and what sort of end users engage in customisation or programming at the moment -- from TiVo to mail rules and smart playlists, to Pipes and Automator. The end user programming community has exploded, but it is still unclear how many 'lay users' utilise such tools. There are clear benefits for an end user to write rules to, say, filter mail, or to perform any other repetitive task. However, the majority of users do not do this. Why? Because the rule-making ability is not known? Because it is too complex? These questions are critical to understanding how we might design support for the new kinds of interactions that become possible with integrated heterogeneous data that cross public, social and personal boundaries. We are currently planning a study into rule-making in existing systems and applying this to aid designing a new interaction with such tools.

Towards end-user understandable user models (scrutability and provenance): Kay highlights [4] that as user models adapt interfaces, the user needs to feel in control and understand what has been adapted, and more importantly, correct or redefine preferences or attributes that were wrong or have changed. Trust is also an element of end-user acceptability and the user's willingness to impart any responsibility to an autonomous entity. Scrutability is one solution to this.

Addressing information overload: although our aim is to reduce time and effort on behalf of the user, part of the value of structure in blending public and personal data and the context that allows is the ability to explore relationships and similarities, to pursue avenues that the user did not know about before but may be deemed relevant -- the gig recommendation in the earlier scenario, for instance. Part of this work is not just about performing repetitive tasks, or suggesting new things of interest, but also about cutting down those things to a manageable level, or saying 'there is a gig

[6] http://www3.ietf.org/proceedings/02mar/I-D/draft-ietf-calsch-many-xcal-01.txt
[7] http://www.w3.org/2003/01/geo/

coming up tonight, but you've just been working for 48 hours solid on a paper deadline, so maybe you should just rest instead.' It is as much about determining what not to show, versus what to show, and how to let the user know that (scrutability as mentioned before) and the interaction in altering those preferences.

Privacy: is clearly an issue in dealing with multiple social/personal data sources, though by conducting *client*-side modeling, we can blend the personal and public on the user's own PC, without personal data being released.

Conclusion

We propose that the Semantic Web on the desktop provides new opportunities in interaction design challenges on at least three levels. First, it enables automatic association of a variety of contexts with information as it is captured and used, providing new opportunities for exploration of personal resources. Such context rich association may also open up opportunities for new kinds of representations to be associated with data to assist in knowledge building and querying. Second, our exploration with AtomsMasher suggests a little semantics enables public, social and personal data to be blended according to behaviours to be associated with these contexts, resulting in great potential value (with challenges in user interface/interaction). Third, these blends enable what were once considered impossible automated data management possibilities to become a reality: our work here so far shows how important not only the Semantic Web, but interaction design is to the success of these approaches. Critically, these early explorations demonstrate that the blend of the Semantic Web with user interaction, the SWUI agenda taken to the desktop, may indeed cause the desktop to disappear as the primary interaction metaphor for computing, and foster a new paradigm for personal computing interaction. We have sketched out several approaches that need considerable exploration in terms of reducing the load on the user to design auto-interactive behaviours that are relevant to them, and look forward to exploring these further with the SWUI community.

References

[1] Adar, E. Karger, D. & Stein, L. A. *Haystack: per-user information environments*. CIKM '99.

[2] Berners-Lee, T.; Hendler, J. & Lassila, O. (2001), 'The semantic Web', *Scientific American* 284(5), 28--37.

[3] Ducheneaut, N. and Bellotti, V. E-mail as habitat: an exploration of embedded personal information management. *interactions* 8, 5 , 2001, 30-38

[4] Kay, J. (2001), 'Scrutability for personalised interfaces', ERCIM NEWS, Special Theme Issue on Human Computer Interaction 46, 49--50.

[5] schraefel, m.c., André, P. & Van Kleek, M. *This Time It's Personal: from PIM to the Perfect Digital Assistant*. CHI2008 PIM Workshop.

[6] schraefel, m.c., Wilson, M., Russell, A., & Smith, D.A. (2006). mSpace: improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM* 49(4), 47--49.

[7] Van Kleek, M. & Shrobe, H. (2007), A Practical Activity Capture Framework for Personal, Lifetime User Modeling, *User Modeling*.

[8] Van Kleek, M., André, P., Smith, D.A., Wilson, M.L., Bernstein, M., Karger, D., & schraefel, m.c. (2007). *AtomsMasher: PeRSSonalized Information Delivery and Management on the Web*. Technical Report: http://eprints.ecs.soton.ac.uk/14827

[9] Van Kleek, M., Bernstein, M., Karger, D., and schraefel, m.c*. Gui -- phooey!: the case for text input*. UIST 2007.

[10] White, R., Kules, B., Drucker, S. & schraefel, m.c (2006), Supporting Exploratory Search, Introduction, Special Issue, *Comm. of the ACM* 49(4), 36--39.